

Compte rendu TP Atelier 07 : TD5 :
Responsive Design

Sommaire

Introduction	1
Exercice 2	1
Exercice 3	2
Exercice 4	2
Exercice 5	3
Exercice 6	3
Exercice 7	4
Exercice 8	4
Exercice 9	4
Conclusion	6

Introduction

Ce TP avait pour objectif d'apprendre à créer des designs responsives, c'est-à-dire adapter un site web pour qu'il s'affiche correctement sur tous les appareils (ordinateur, tablette, téléphone) tout en restant esthétique et accessible.

Exercice 2

J'ai découvert comment utiliser les propriétés `max-width` et `min-width` pour contrôler la taille des éléments en fonction de l'écran, empêchant ainsi qu'ils ne deviennent trop grands ou trop petits.

```
max-width: 500px;  
min-width: 200px;
```

Exercice 3

J'ai testé l'affichage du site sur différentes résolutions d'écran pour vérifier son adaptation et identifier les problèmes d'affichage sur mobile.

```
Dimensions: 555 x 852  
Responsive: 100%  
No throttling  
Save-Data: default
```

Exercice 4

J'ai exploré les propriétés `flex-grow`, `flex-shrink` et `flex-basis` pour contrôler le comportement des éléments dans un conteneur flex, permettant une adaptation fluide selon l'espace disponible.

```
article {  
  flex-grow: 3;  
  flex-shrink: 2;  
  flex-basis: 300px;  
}  
  
aside {  
  flex-grow: 1;  
  flex-shrink: 1;  
  flex-basis: 200px;  
}  
  
main {  
  display: flex;  
  width: 100%;  
  gap: 20px;  
}
```

Exercice 5

J'ai utilisé les outils de développement pour simuler l'affichage sur un Samsung Galaxy S8+, ce qui m'a permis de visualiser et corriger les problèmes spécifiques aux mobiles.

```
Dimensions: Samsung Galaxy S8+  
360 x 740  
100% No throttling  
Save-Data: default
```

Exercice 6

J'ai ajouté la balise `<meta name="viewport">` pour désactiver les ajustements automatiques du navigateur et prendre le contrôle total de l'adaptation sur mobile.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Exercice 7

Certaines fonctionnalités testées ne fonctionnaient pas comme prévu, ce qui m'a obligé à rechercher des solutions alternatives pour résoudre ces problèmes.

Exercice 8

J'ai appris à utiliser les media queries pour appliquer des styles CSS spécifiques selon la taille de l'écran, comme masquer un tableau ou modifier la disposition des éléments sur mobile.

```
@media screen and (max-width: 768px) {  
  table {  
    display: none;  
  }  
}  
  
@media screen and (max-width: 480px) {  
  main {  
    flex-direction: column;  
  }  
}
```

Exercice 9

J'ai créé un menu burger interactif qui remplace le menu classique sur les petits écrans, avec une animation pour le faire apparaître et disparaître au clic.

```
<div class="burger">
  
  <div id="menu2">
    <div><a href="./index.html">Accueil</a></div>
    <div><a href="./facts.html">Facts</a></div>
    <div><a href="./news.html">Actualités</a></div>
    <div><a href="./contact.html">Contact</a></div>
  </div>
</div>
```

```
.burger img {
  position: fixed;
  top: 10px;
  right: 20px;
  width: 50px;
  cursor: pointer;
  z-index: 20;
}

#menu2 {
  position: fixed;
  top: 0;
  right: 0;
  width: 80vw;
  height: 100vh;
  background-color: rgba(91, 189, 191, 0.95);
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  gap: 20px;
  z-index: 10;
  margin-right: -80vw;
}

.burger:hover #menu2 {
  margin-right: 0;
}
```

Conclusion

Ce TP m'a permis de maîtriser les bases du responsive design. J'ai trouvé la partie sur les flexbox assez complexe initialement, mais les media queries se sont révélées plus intuitives. Globalement, j'ai acquis les compétences nécessaires pour adapter des sites web à tous types d'appareils.